|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №3**

Выполнил студент группы ИНБО-02-20 Резаев А.А.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

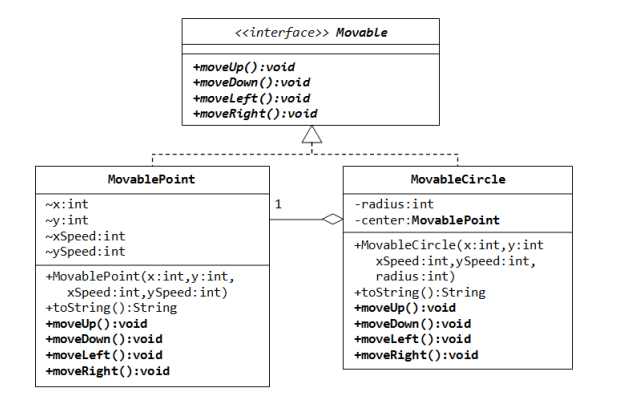
Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

## 1)Переписать суперкласс Shape и его подклассы так как это представлено на диаграмме Circle, Rectangle and Square.



2) Нужно написать два класса MovablePoint и MovableCircle - которые реализуют интерфейс Movable . 

## **Реализация заданий: задание 1**

Листинг 1.1 – класс Shape

|  |
| --- |
| package PR\_3;  public abstract class Shape{  protected String color;  protected boolean filled;  protected double area;  protected double perimeter;  public Shape() {  }  public Shape(String color, boolean filled) {  this.color = color;  this.filled = filled;  }  public String getColor() {  return color;  }  public void setColor(String color) {  this.color = color;  }  public boolean isFilled() {  return filled;  }  public void setFilled(boolean filled) {  this.filled = filled;  }  public String toString() {  return toString();  }  public abstract double getArea();  public abstract double getPerimeter();  } |

Листинг 1.2 – класс Circle

|  |
| --- |
| package PR\_3;  import java.math.\*;  public class Circle extends Shape{  protected double radius;  public Circle(){}  public Circle (double radius){  this.radius=radius;  }  public Circle (double radius, String color, boolean filled){  this.radius=radius;  this.color=color;  this.filled=filled;  }  public double getRadius(){  return radius;  }  public void setRadius(double radius)  {  this.radius=radius;  }  public double getArea(){  return Math.PI\*radius\*radius;  }  public double getPerimeter(){  return 2\*Math.PI\*radius;  }  public String toString(){  return "Shape: circle, radius: "+this.radius+", color: "+this.color+", filled: "+this.filled;  }  } |

Листинг 1.3 – класс Rectangle

|  |
| --- |
| package PR\_3;  public class Rectangle extends Shape{  protected double width;  protected double length;  public Rectangle(){}  public Rectangle(double width, double length){  this.width=width;  this.length=length;  }  public Rectangle(double width, double length, String color, boolean filled){  this.width=width;  this.length=length;  this.color=color;  this.filled=filled;  }  public double getWidth(){  return width;  }  public void setWidth(double width){  this.width=width;  }  public double getLength(){  return length;  }  public void setLength(double length){  this.length=length;  }  public double getArea() {  return width\*length;  }  public double getPerimeter() {  return 2\*(width+length);  }  public String toString(){  return "Shape: rectangle, width: "+this.width+", length: "+this.length;  }  } |

Листинг 1.4- класс Square

|  |
| --- |
| package PR\_3;  public class Square extends Rectangle{  private double side;  public Square(){  this.side = 4;  this.color = "green";  this.filled = true;  }  public Square(double side){  this.side = side;  }  public Square(double side, String color, boolean filled){  this.side = side;  this.color = color;  this.filled = filled;  }  double getSide(){  return side;  }  void setSide(double side){  this.side = side;  }  public void setWidth(double side){  this.side = side;  }  public void setLength(double side){  this.side = side;  }  public String toString(){  return "Shape: square, side: "+this.side+", color: "+this.color+", filled: "+this.filled;  }  } |

## **Задание 2**

Листинг 2.1- интерфейс Movabale

|  |
| --- |
| package PR\_3;  public interface Movable {  public void moveUp();  public void moveDown();  public void moveLeft();  public void moveRight();  } |

Листинг 2.2- класс MovableCircle

|  |
| --- |
| package PR\_3;  public class MovableCircle implements Movable {  private int radius;  private MovablePoint center;  private int x;  private int y;  private int xSpeed;  private int ySpeed;  public MovableCircle(int x, int y, int xSpeed, int ySpeed){  this.x = x;  this.y = y;  this.xSpeed = xSpeed;  this.ySpeed = ySpeed;  }  public String toString(){  return "(x,y): ("+this.x+","+this.y+") speed: ("+xSpeed+","+ySpeed+")";  }  public void moveUp() {  moveUp();  }  public void moveDown() {  moveDown();  }  public void moveLeft() {  moveLeft();  }  public void moveRight() {  moveRight();  }  } |

Листинг 2.3- класс MovableRectangle

|  |
| --- |
| package PR\_3;  public class MovableRectangle implements Movable{  private int x1,x2;  private int y1,y2;  private int xSpeed;  private int ySpeed;  private MovablePoint topLeft;  private MovablePoint bottomRight;  public MovableRectangle(int x1,int y1, int x2, int y2, int xSpeed, int ySpeed){  this.x1 = x1;  this.y1 = y1;  this.x2 = x2;  this.y2= y2;  this.xSpeed = xSpeed;  this.ySpeed = ySpeed;  }  public String toString(){  return "(x1,y1): ("+this.x1+","+this.x1+") (x2,y2): ("+this.x2+","+this.y2+") xSpeed: "+this.xSpeed+" ySpeed: "+this.ySpeed;  }  public void moveUp() {  moveUp();  }  public void moveDown() {  moveDown();  }  public void moveLeft() {  moveLeft();  }  public void moveRight() {  moveRight();  }  } |

Листинг 2.4- класс MovablePoint

|  |
| --- |
| package PR\_3;  public class MovablePoint implements Movable {  private int x;  private int y;  private int xSpeed;  private int ySpeed;  public MovablePoint(int x, int y, int xSpeed, int ySpeed){  this.x = x;  this.y = y;  this.xSpeed = xSpeed;  this.ySpeed = ySpeed;  }  public String toString(){  return "(x,y): ("+this.x+","+this.y+") speed: ("+xSpeed+","+ySpeed+")";  }  public void moveUp() {  moveUp();  }  public void moveDown() {  moveDown();  }  public void moveLeft() {  moveLeft();  }  public void moveRight() {  moveRight();  }  } |

## **Вывод**

Получены знания по работе с абстрактным классом.